



A tabu search heuristic for a dynamic transportation problem of patients between care units

Yannick Kergosien, Christophe Lenté, D. Piton, Jean-Charles Billaut

► To cite this version:

Yannick Kergosien, Christophe Lenté, D. Piton, Jean-Charles Billaut. A tabu search heuristic for a dynamic transportation problem of patients between care units. 2010. hal-00488270

HAL Id: hal-00488270

<https://hal.science/hal-00488270>

Preprint submitted on 1 Jun 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A tabu search heuristic for a dynamic transportation problem of patients between care units

Y. Kergosien¹, Ch. Lenté¹, D. Piton² and J-C. Billaut¹

¹Université François Rabelais Tours

Laboratoire Informatique, 64, av. Jean Portalis, 37200 Tours.

email: {yannick.kergosien|christophe.lente|jean-charles.billaut}@univ-tours.fr

²Centrale des ambulanciers, CHRU TOURS, 37044 Tours.

email: d.piton@chu-tours.fr

May 11, 2010

Abstract - The ambulance central station of the Hospital Complex of Tours (France) has to plan the transportation demands between care units which require a vehicle. Some demands are known in advance and the others arrive dynamically. Each demand requires a specific type of vehicle and a vehicle can transport only one person at a time. Moreover, transportations are subject to particular constraints: priority of urgent demands, disinfection of a vehicle after the transportation of a patient with contagious disease, respect of the type of vehicle, etc. This problem is related to the *Dial A Ride Problem*. For solving the dynamic version of this problem, we propose a tabu search algorithm inspired by [Gendreau *et al.*, 1999]. This method supports an adaptive memory which stores routes and consists in running a tabu search algorithm several times: one for improving the set of initial solutions, one for the neighborhood exploration and finally for improving the final solution. Computational experiments show that the method can provide high-quality solutions for this dynamic transportation problem.

Keywords: Transportation, Real-time, Health care, tabu search heuristic, Vehicle routing

1 Introduction

The Hospital Complex of Tours (France) is dispatched over several geographical service areas with different specialties of care units (see Fig. 1). It often occurs that a patient has to be transferred to another hospital or to another care unit for specific medical examinations, operating theater use, magnetic resonance imaging use, etc. When the transportation is necessary within the same building, the care staff in the original service of the patient provides this transportation independently. In this case, no specific vehicle is required. When the transportation needs a vehicle, the response to the transportation demand is given by the operators of the *ambulance central station*. The demands differ in the type of vehicle needed and in their priorities. Some demands are known in advance and the others arrive dynamically. The purpose of this paper is to propose a method to help the operators satisfying in real time all the transportation demands (and constraints) so that the cost for the hospital is minimized and so that the demands are satisfied on time or as quickly as possible [Kergosien *et al.*, 2008].

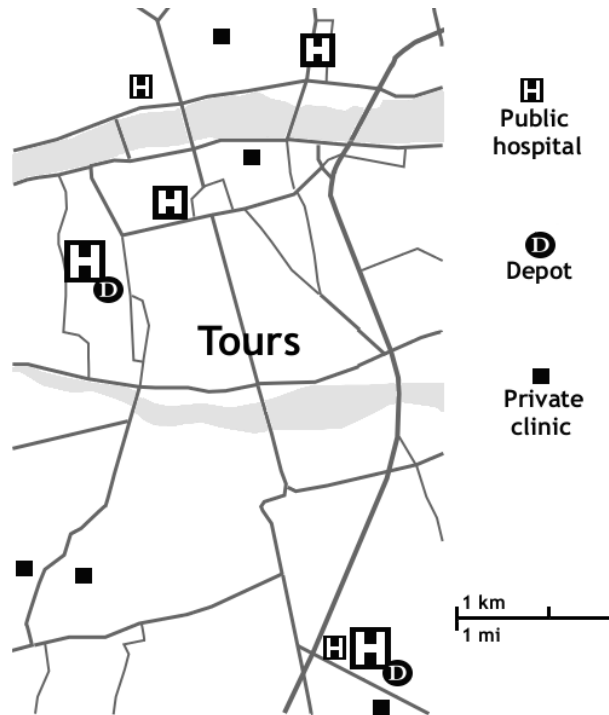


Figure 1: Map of Tours and position of hospital units

The paper is organized as follows. In Section 2, we describe the problem and the context of the study. Section 3 contains a literature review related to this problem. In Section 4, we propose a modeling of a solution and of the resolution method applied by the operators of the ambulance

central station. For solving the dynamic version of the problem, a tabu search algorithm with an adaptive memory is described in Section 5. Computational experiments are reported in Section 6. Finally, we conclude the study and propose some further research directions.

2 Problem description

The ambulance central station of the Hospital Complex of Tours is composed by 25 vehicles of different types and 55 ambulance drivers. More than 46,000 transportations are performed every year, with a mean value of 130 daily transportations. The demands are planned dynamically by one or two persons, called operators of the ambulance central station. In real time, they receive new demands and they have to assign a vehicle to each one, considering vehicle availabilities, time constraints, geographical situations, required equipment and the set of known demands that they will have to satisfy in a near future. They also have the possibility to seek the help of private ambulance companies, if necessary. Each transportation demand generates a cost related to the total distance and the use of a private company vehicle generates an additional cost.

We assume in our study that the crews (two ambulance men per ambulance) are already constituted and that their timetables are fixed, which is true in practice. The number of crews working simultaneously varies from 1 to 9 during the day. We consider two types of vehicles: vehicles of type *C* are classical ambulances that can ensure the major part of the demands and vehicles of type *A* are heavy ambulances for delicate transportations. Vehicles of type *A* are larger than those of type *C* and can accommodate a complete health staff and more medical equipment. For this reason, they are used for sensitive transportations.

Transportation demands arrive from different services of the hospital. In average, only 30% of the demands are known at the beginning of the day. The other demands arrive dynamically. We associate to each demand a priority and a forecasted starting time that has to be respected. Transportation demands issued from the *Emergency Medical Service* (EMS) have a very high priority level and require a short response time. Operators have to react immediatly. Three types of transportation demands are considered: usual, transportation demands requiring a medical monitoring and contagious transportation demands. Transportation demands with medical monitoring require the presence of a physician and contagious transportation demands require the disinfection of the vehicle after the transportation.

3 Literature review

The problem considered in this paper is related to the *Dial A Ride Problem* (DARP), which is a particular case of *Vehicle Routing Problem with Pickup and Delivery* (VRP-PD) [Desaulniers *et al.*, 2002]. One of the first studies of this problem is given in [Stein, 1978]. VRP-PD is related to transportation of goods whereas DARP deals with the transportation of people, a specificity being that a person has a unique departure point and a unique destination.

One of the first studies about DARP is presented in [Psaraftis, 1980] for the case of one vehicle and in [Jaw *et al.*, 1986] for the case of multiple vehicles. An extensive state-of-the-art survey of models and algorithms for DARP can be found in [Cordeau and Laporte, 2007]. The case of a heterogeneous fleet is treated in a few papers. In [Dumas *et al.*, 1989], [Desrosiers *et al.*, 1991] and [Ioachim *et al.*, 1995], the authors propose two-stage heuristics based on a grouping technique of demands and a routing procedure. In [Borndörfer *et al.*, 1997] the authors study the same problem applied to the transportation of disabled persons. They build a two-step heuristic approach. In [Melachrinoudis *et al.*, 2007], the authors consider the DARP in a context of health care organization. In this problem, the same vehicle is used twice for the same patient: for driving the patient to the hospital and for driving the patient back to the starting point. The authors propose an integer programming formulation and a tabu search algorithm.

Only the static version of the problem is considered in the previous papers. In [Ichoua *et al.*, 2007], the authors review the literature on dynamic vehicle routing problem. They present an adaptation of a static algorithm, discuss the issue of diverting and propose some techniques to anticipate the future demands. [Berbeglia *et al.*, 2010] presents a literature review of dynamic pickup and delivery problems and a part of the paper is dedicated to the dynamic DARP. Among real-life studies, [Madsen *et al.*, 1995] are interested in the dynamic case of transportation of disabled people. In addition to constraints of time windows and a heterogeneous fleet, constraints of multi-dimensional capacities and priorities of customers are taken into account. The authors propose an algorithm based on an insertion heuristics, called REBUS, to solve the problem in a dynamic environment with a short computation time. In [Coslovich *et al.*, 2005], the authors consider a special case where the demands are emitted on-line by customers at the station point of the vehicles. The driver of the vehicle accepts or declines the demand. The authors propose a two-phase solution method. The first phase consists in testing the insertion of a new customer in the current route so that the driver can answer this demand quickly. The second phase, which is run off-line when

the vehicle moves between two successive stops, is aimed at creating a feasible neighborhood of the current route. Another case with several heterogeneous vehicles based on a unique depot is tackled in [Teodorovic and Radivojevic, 1998]. When new demands occur, the first phase consists in assigning each demand to a vehicle. In a second phase the schedule of the selected route is optimized. Both algorithms are based on fuzzy logic. A multimodal transportation problem of people is considered in [Horn *et al.*, 2002]. A planning system called L2sched is proposed for vehicle routing with a heterogeneous fleet of vehicles. When a new request occurs, it is inserted in a current route using a minimum cost insertion scheme. The routes are also improved thanks to a local search. A more general case of the DARP is presented in [Attanasio *et al.*, 2004]. The objective of the problem is to maximize the total number of accepted demands arriving on-line. For solving this problem, the authors compare several modes of parallelization of a tabu search inspired by [Cordeau and Laporte, 2003]. A final study on dynamic DARP is made in [Xiang *et al.*, 2008]. The authors consider several types of possible events (new demands, cancellation of demands, vehicles breakdown, etc.). For solving this problem, the authors built an algorithm based on a static case of the problem studied in [Xiang *et al.*, 2006].

One of the main differences between DARP and our problem is the vehicles capacity, limited to one person in our problem. The reason is that patients have to be transported in lying position on a stretcher and ambulances can contain only one stretcher at a time. So, when a vehicle takes a patient at his/her starting point, the vehicle has to go directly to his/her destination. In [Laporte, 1997], the author proposes a general method for transforming a transportation problem with unit capacity into a traveling salesman problem. The other differences with DARP are some characteristics of the transports, due to some hospital constraints described later. Another difference with the DARP is about the subcontracted demands. Unlike most of the DARP, all vehicles must not necessarily cover all the transportation demands.

In the field of the vehicle routing problems in hospital context, another research area is the problem of emergency vehicle location (see [Alsalloum and Rand, 2006], [Araz *et al.*, 2007] or [Rajagopalan *et al.*, 2008]). The problem is to optimize the covering zone for emergency services. In [Shang and Carolyn, 1996], the authors consider the transportation of medical artefacts, mostly medical records, which is similar to a VRP-PD with time windows and without capacity constraints. Finally in [Beaudry *et al.*, 2008] the authors develop a heuristic for a dynamic DARP. The objective is to insert new demands in existing routes respecting some constraints such as patient isolation or the requirement of medical items. The authors present a heuristic by insertions plus a tabu search

algorithm. The authors deal with the problem in the most possible generic way so as to be able to apply it to other cases. This problem is very close to ours. However, in our problem there is no transportation by wheelchair but only by stretcher, with only one patient per vehicle. In addition, crews can change their vehicle at any time (to perform a transportation with medical monitoring after disinfection). Finally, among the events that may happen, the case of crew suppression is possible. This case arises when the EMS must transport an emergency patient to another city. A crew is then unavailable for a long time. In [Hanne *et al.*, 2009] the problem is more applied to the real case than in [Beaudry *et al.*, 2008]. The work carried out in collaboration with the central *patient transportation department* (PTD). The PTD manages intra-hospital transportation and inter-building transportation by vehicles (with several patients per vehicle, and different vehicle configurations) or by foot (gurneys, beds, or wheelchairs). The demands have quite the same characteristics as in [Beaudry *et al.*, 2008] (time windows, modes of transportation, required equipment, etc.) in a dynamic environment. A solution is evaluated according to the patient inconvenience and the hospital cost. A discrete-event simulation model is proposed to reproduce the dynamic environment of transportations, by using real data of the PTD. Then, a transportation planning system called Opti-TRANS is presented. This system allows to book, schedule, dispatch, monitor and report the transportation demands. Several optimization algorithms have been combined: two methods for demand assignment, three methods for routing and scheduling decisions and an evolutionary algorithm. Thanks to Opti-TRANS, the authors notice an improvement of patient satisfaction and a reduction of the transportation cost. They indicate that this system can also be used for other types of transportation (hospital logistics of material). The differences between our study and this study are the following. We consider one patient per vehicle. Vehicles are not directly related to the crews because a crew can change his vehicle according to the type of transportation they have to perform (e.g. a vehicle disinfection is not necessarily done just after the contagious transport: a crew can change his vehicle and come back later for performing this operation). Moreover, since the number of crews is not sufficient in comparison with the number of demands, the main objective leads to subcontract the least possible demands.

The solution method presented in this paper is based on [Gendreau *et al.*, 1999], where the problem is based on a static version of the problem described in [Taillard *et al.*, 1997]. The considered problem is a static vehicle routing problem with soft time windows (noted VRP-STW): given a set of customers to serve and a set of vehicles placed in a unique depot, determine the route of each vehicle in order to satisfy the demands and to minimize the total distance traveled

plus a lateness penalty. In [Gendreau *et al.*, 1999], all customers are not known in advance, some of them reach on-line. From the original version of the algorithm, a parallel implementation based on [Badeau *et al.*, 1997] is used to take account of the dynamic environment. The main idea for interacting with the environment when no event occurs is to optimize the routes by using a tabu search. When an event occurs, the following steps are applied: stop the optimization, update the adaptive memory and solutions depending on the treated event and restart the optimization. Only two events are considered: arrival of a new customer or end of service at a customer destination. In [Ichoua *et al.*, 2000], the authors take this method and add diversion issues. These diversions allow to improve the results. The improvement consists in diverting a vehicle away from its planned destination to serve a close customer location, that has just occurred, unlike the previous studies assume that the next destination of a vehicle is fixed. In our problem, we also allow diversions. Another extension is presented in [Gendreau *et al.*, 2006]. In this extension, the customer is represented by a demand that must be carried from a pick-up location to a delivery location. The problem is a typical courier service application with a dynamic environment.

In this paper, we adapt the static method of [Taillard *et al.*, 1997], like in [Gendreau *et al.*, 1999]. Our problem differs from these problems by several aspects: we have not a unique depot, but patients to move between two points with several depots; transportation of patients may need several types of vehicles, some of them cannot be reused immediately (disinfection); hard time windows have to be respected (no tardiness allowed). We also have an additional difference with most of the vehicle routing problems in the literature: the opportunity to subcontract demands. Therefore, serving all the customers is not a constraint.

4 Modeling

4.1 Notation and hypotheses

The data that we consider are the following.

- A set of crews, noted \mathcal{M} , where a crew is composed of two ambulance men assigned to a vehicle. Each crew $m \in \mathcal{M}$ is characterized by:
 - the date of availability, that corresponds to the beginning of the timetable of the crew,
 - a depot D_m where the crew starts and comes back at the end of the working day,
 - the date when the crew must be back to the depot (corresponds to the end of the working

day). Because this is not really a hard constraint, the solutions which induce an overtime work are penalized by a cost. The function $e^{(r_m - T)}$ has been chosen in order to limit as much as possible the overtime work, with r_m the tardiness of crew m and T the time during which this overtime work is acceptable.

- A set of transportation demands, denoted \mathcal{N} . Each demand $i \in \mathcal{N}$ is characterized by:
 - the starting point s_i , which corresponds to the initial care unit of the patient,
 - the destination of the patient, denoted f_i ,
 - the type of transportation, among {classic, with medical monitoring, contagious},
 - a priority, denoted u_i : priority 1 corresponds to the most urgent transportation demand. In particular, the demands of the Emergency Medical Assistance Service have priority 1,
 - the traveling time $t(a, b)$ from location a to location b . $t(s_i, f_i)$ is the travel time from s_i to f_i .
 - the time needed to handle the patient outside of the vehicle (from his bed to the vehicle, then from the vehicle to his destination in the unit care), noted p_i^{care} ,
 - the date at which the transportation has to start, noted e_i . We also associate a time window $[e_i, e_i + \alpha_i]$ during which the demand has to be satisfied. This time window depends on the priority of the demand. The higher the priority, the smaller the time window. We set $\alpha_i = \tau_0 + \delta \times u_i$, with u_i the priority defined previously, τ_0 the value of the maximum tardiness allowed for the more urgent demand ($\tau_0 = 5$ minutes in the following) and δ the difference in minutes between two priority levels ($\delta = 5$ min).
 - a cost of subcontracting, cs_i . This cost depends on the demand and not on the private company. The expression is $cs_i = 1,5 \times t(s_i, f_i) + 60$ plus a constant additional cost if a vehicle of type A is used or if a disinfection is needed. We assume in the following that this cost is strictly greater than the cost for transportation i performed by the Hospital. We call “*unsatisfied demand*” a transportation demand that is assigned to a private company, and not to the hospital.
- A set of vehicles \mathcal{V} , where each vehicle $v \in \mathcal{V}$ is characterized by:
 - a type: C for classical ambulance, A for medicalized ambulance, with a limited number of vehicles of each type.

- a cost $c_{i,j}^v$ if vehicle v performs demand i and then goes to the starting point of demand j . The cost is calculated according to the kilometric costs, and the expression is $c_{i,j}^v = 0,2 \times (t(s_i, f_i) + t(f_i, s_j))$ plus an additional cost if a vehicle of type A is used.
- a depot.

When a crew has to perform demand i and then j , they have implicitly to drive from the destination of demand i to the starting point of j . It is assumed that the travel time and the cost of this travel are known. Moreover, the time needed for the change of vehicle (p_{ch}), the time to drop a physician off (p_{phy}) and the time of disinfection (p_{dis}) are not neglected and are assumed to be constant.

Before the physician returns to his/her care unit, the transportation with medical monitoring can pick up another patient, but only if this operation is carried on rather quickly. This constraint is modeled by limiting the duration of the return of the physician. If the transportation with medical monitoring completes at time t_1 , then the physician has to be back to his care unit before date $t_1 + p_{return} + p_{max}$, with p_{return} the duration of his/her direct return (equal to $t(f_i, s_i)$ if i is the transportation demand) and p_{max} the duration of his/her accepted detour (constant duration fixed to 15 min).

The vehicles have only a capacity of one patient and special constraints have to be respected according to the type of transportation demand:

- *classic* transportation demands can be performed by a vehicle of type A or C .
- transportation demands with medical monitoring have to be performed by a vehicle of type A and need the presence of a physician to keep the patient under surveillance. This physician comes from the care unit of the patient and has to be driven back to this care unit by the same ambulance after the transportation of the patient. This return can be executed with another patient on board who has to go to this unit.
- *contagious* transportation demands are performed by a vehicle of type C . The vehicle has to be disinfected before another use, immediately after the travel or after the crew performs another transportation activity with another vehicle (disinfection takes around one hour).

For a dynamic use, the method also needs the state of the vehicles ('in use', 'to be disinfected' or 'available') and their locations:

- a date of availability that corresponds either to the completion time of the current activity performed by the crew (transportation activity, break period, disinfection, etc.) or to the current date if they are not performing an activity,
- a location or starting point: position of the crew at its date of availability.

The problem is to assign the transportation demands to the crews of the hospital and to determine the routes of the vehicles. The objective function to minimize is the cost of transportation performed by the hospital plus the cost of transportation subcontracted to the private companies plus tardiness penalties.

4.2 Modeling of a solution

A solution at time t is given by a set of routes with one route per crew. A route is a chain of vertices, defined as follows:

- vertex $*$ is the position of the crew at the end of the current activity or the current position if they are not performing an activity,
- activity of performing a transportation demand,
- activity of changing a vehicle at a depot for transportation demands with medical monitoring or contagious transportation demands,
- activity of disinfecting a vehicle at a depot. These vertices are inserted after the contagious transportation demands,
- activity of driving back a physician to his care unit. These vertices are inserted after the transportation demands with medical monitoring,
- vertex D_m , position of the depot of crew m .

We note $d(i, j)$ the length of edge (i, j) . This length represents the time interval between the starting time of activity i and the starting time of activity j . We give below the computation details for the most important cases:

- if i and j are two transportation demands: $d(i, j) = p_i^{care} + t(s_i, f_i) + t(f_i, s_j)$ (see Fig. 2).
- if i is a transportation demand and $j = D$ a depot (in the case of a disinfection or of a change of vehicle or at the end of the working day): $d(i, j) = p_i^{care} + t(s_i, f_i) + t(f_i, D)$.

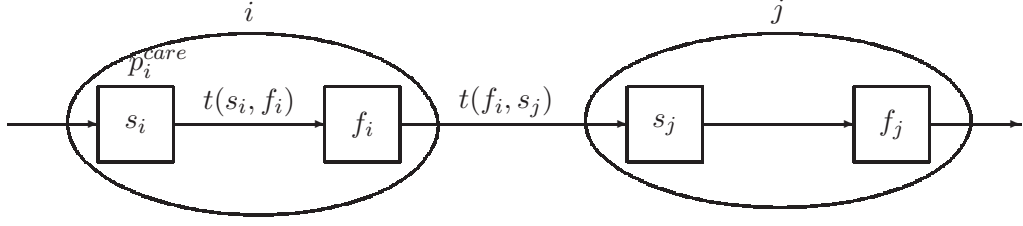


Figure 2: Length of edge (i, j) if i and j are two transportation demands

- if $i = D$ is a depot and j a transportation demand: $d(i, j) = p + t(D, s_j)$ ($p = p_{ch}$ for changing a vehicle and $p = p_{dis}$ for disinfection).
- if i is a transportation demand and j a physician located in s_i : $d(i, j) = p_i^{care} + t(s_i, f_i) + t(f_i, s_i)$. If the physician is located in s_u , the expression of $d(i, j)$ is modified accordingly.

At time $t = 0$, the first and the last vertices of a route are identical: each crew and each vehicle must come back to their starting point at the end of the working day. At any time $t > 0$, the first vertex of the chain corresponds to the current position of the crew at time t .

Example

An example with two crews, four care units H_1 to H_4 and a single depot D located at H_1 is given in Figure 3. We consider five transportation demands $\mathcal{N} = \{a, b, c, d, e\}$, where $\{a, b, c\}$ are classical transportation demands, $\{d\}$ requires a medical monitoring and $\{e\}$ a contagious transportation demand.

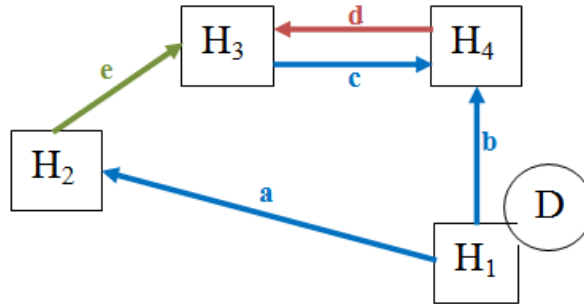


Figure 3: Example with five transportation demands

Figures 4 and 5 show an example of a solution modeling. The vertex $*$ represents the current position of the crew at time t .

In this example (Fig. 4), crew $m = 1$ starts to perform the transportation demand e , then goes to the depot D for changing its vehicle of type C to another vehicle (because of contagiousness),

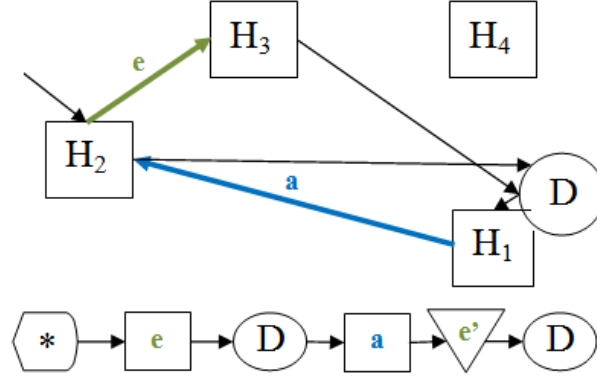


Figure 4: Route of crew 1

then performs demand a and comes back to the depot for the disinfection of the previous vehicle (activity e'). We assume here that if the disinfection of the vehicle was done immediately after demand e , demand a would not be performed on time.

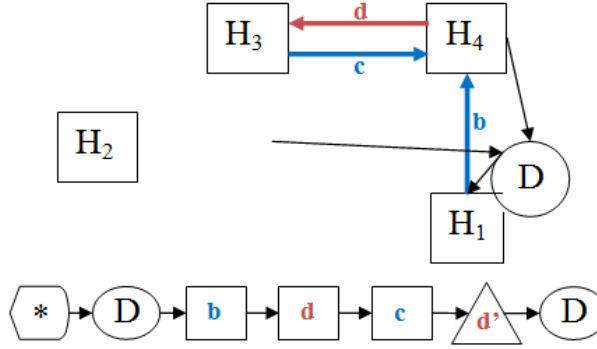


Figure 5: Route of crew 2

Crew $m = 2$ goes to the depot D for changing its vehicle of type C into a vehicle of type A , then performs demand b and the transportation with medical monitoring d . After this, the crew has to drive back the doctor (demand d' which has been inserted) but the crew can take another patient at the same time (demand c). At the end, the crew returns to the depot to recover its vehicle of type C .

4.3 Modeling of the operator assignment rule

The behavior of the operator can be modeled by the following algorithm.

At the beginning of the day, the set of the first demands to satisfy are assigned to the closest crews and vehicles. If some demands of this set cannot be assigned to the vehicles of the hospital,

these demands are subcontracted.

Then, by anticipation, the operator assigns to each vehicle the next two or three transportation demands. By this way, both the unexpected and the known demands are gradually integrated into the planning of crews and vehicles. The rule consists in assigning a transportation demand to the vehicle that can be available as soon as possible for performing the demand. The constraints like the time needed for disinfection or for a vehicle change are also taken into account. Again, the demands that cannot be performed by the vehicles of the hospital are subcontracted.

The operator behavior has been implemented as a primary heuristic method, noted *DynOper*.

5 Solution method

5.1 General algorithm

Because our problem differs from the problems presented in [Taillard *et al.*, 1997] and [Gendreau *et al.*, 1999], the general method has been adapted and some structural elements have been changed for obtaining high-quality solutions. The coding of a solution has been extended in order to take account of the different activities that can perform a team. Some steps have been adapted, like the upgrade of routes when a new demand occurs, the neighborhood operator of the tabu search, etc. Other steps have been added for subcontracted demands, like a phase for the insertion in an existing route of unassigned transportation demands.

The main idea of the algorithm is the following. A list of feasible routes is stored in an “*adaptive memory*”. These routes come from high quality solutions previously built. When a new transportation demand arrives at time t , this list is updated: the transportation demands already performed are deleted, the positions of the crews and their availability dates are modified. The new transportation demand is not inserted at this step.

A subset of disjoint routes is extracted from the adaptive memory for giving a solution. Some demands are thus unsatisfied (these demands are assigned to private companies by default). The aim of the next step is to improve this solution. Then, repeated K times, the set of routes is split into two parts (not necessarily disjoint), each part is improved using a tabu search algorithm and the two modified parts are gathered again to create a new set of routes. This solution is improved (by taking the changes of vehicles into account) and some unsatisfied demands are inserted. The best found solution becomes the current one. The routes of this solution are inserted in the adaptive memory, some others are deleted.

This whole process is repeated J times. At the end, the best found solution is improved and the routes are inserted in the adaptive memory. The algorithm is denoted by *Dynamic* (see algorithm 1). t denotes the current time.

Algorithm 1 *Dynamic*

```

1:  $AM(t) \leftarrow \text{update}(AM(t-1))$ 
2: for  $j = 1$  to  $J$  do
3:    $S \leftarrow$  subset of disjoint routes of  $AM(t)$  (some transportation demands may not appear in  $S$ ,
      they constitute the set of unsatisfied demands)
4:   Try to insert all unsatisfied demands in the routes of  $S$ 
5:   for  $k = 1$  to  $K$  do
6:     Decomposition: create two subsets of routes  $S_1$  and  $S_2$  from  $S$ 
7:     Apply the tabu search on  $S_1$  which gives  $S'_1$ 
8:     Apply the tabu search on  $S_2$ , keeping the previous tabu list, which gives  $S'_2$ 
9:     Reconstruction: gather  $S'_1$  and  $S'_2$  and obtain a new solution  $S'$ 
10:     $S \leftarrow S'$ 
11:    Improve  $S$  (reorganization of the changes of vehicles)
12:    Try to insert all unsatisfied demands in the routes of  $S$ 
13:  end for
14:   $S \leftarrow$  the best solution found during the  $K$  iterations
15:  Add the routes of  $S$  to  $AM(t)$ 
16: end for
17: Post optimization of the best solution by using the tabu search and add its routes to  $AM(t)$ 

```

5.2 Management of the Adaptive Memory

The adaptive memory is a set of routes sorted in decreasing quality order. The quality of a route is defined by the objective value of the solution it comes from. Three important actions involve the adaptive memory: updating the routes with new events (line 1), adding the routes of a solution (line 15) and constructing a new solution from the routes in the adaptive memory (line 3).

- At time $t-1$, an adaptive memory coming from the former iteration is denoted by $AM(t-1)$. At time t , a new transportation demand arrives. We have to consider this demand and to update $AM(t-1)$, i.e. to cancel the performed demands, to update the positions of the

crews and their availability dates. For this last update, we must determine the positions of the crews at this time. If a crew is performing a transportation (either of a patient or of a physician), its date of availability is equal to the completion time of this transportation and the final location of this transportation becomes the next starting point of the crew. If a crew does not transport a patient, its date of availability is equal to the current time t and the starting point of the crew is its current position. The crew may be along the way towards the next transportation demand. This method of managing the adaptive memory is in favor of diversion, if a new transportation demand appears in the neighborhood of its current position. If the time t appears after the end of the working day of a crew, this crew is removed. We obtain a new adaptive memory which is used to solve the problem (line 1). We denote $AM(t)$ the adaptive memory obtained at time t .

- The adaptive memory always contains a fixed number of different routes, coming from the best solutions obtained during the search procedure. When new routes are added to the adaptive memory, the worst routes are deleted. A new route is inserted in the memory at the right place, according to its quality. If the same route was already present in the list, but with a weaker quality, this route is deleted and the new route is inserted at the right place.
- The construction of a new solution from the adaptive memory works as follows. A common fixed probability of selection, noted ρ (a parameter of the algorithm), is associated to each route. The routes are considered in the memory order and a route is selected if a randomly generated number is lower than or equal to ρ . After the selection of a route, the other possible routes that have at least one demand in common with the selected route, or those that use a same vehicle at the same time are eliminated for the next selection. If the set of candidate routes is empty and if there are remaining demands, these demands are assigned to vehicles according to the initialization procedure, described below.

5.3 Initialization procedure

At the beginning of the day (time $t = 0$), the ambulance central station is aware of about 30% of the transportations that have to be performed in the day. We initialize the adaptive memory with several random solutions, built as follows. First, the vehicles are sorted randomly for having a wide variety of routes. Then, the demands are treated in an arbitrary order: a demand is assigned to the first vehicle in the list that can take the demand on time and that satisfies the constraints.

When a transportation demand with medical monitoring or contagious is inserted in a route, the vertices representing the changes of vehicles or the disinfection and the return of the physician are also inserted, so that the route remains always feasible. After that, for each initial solution, a tabu search (see section 5.8 for a description) is launched with a small number of iterations (only five after some experiments), for trying to improve the solution. This procedure allows generating a high variety of routes because vehicles selection and demands assignments are treated in an arbitrary order.

5.4 Improving a solution

For improving a solution, two steps have been implemented: ‘insertion of unsatisfied demands’ (lines 4 and 12), and ‘reorganization of the changes of vehicles’ (line 11).

After the construction of a solution from the adaptive memory, some transportation demands may remain unsatisfied. By default, these demands are subcontracted to private ambulance companies and generate a cost. To improve the solution, these demands are inserted into the current solution. The procedure consists in considering the demands one by one, and testing if they can be inserted in one of the routes of the solution, provided that this insertion reduces the cost of the solution. When a transportation demand with medical monitoring is inserted, the procedure also inserts the vertices corresponding to the change of vehicle (type A) and the vertex corresponding to the return of the physician such as the route remains feasible. The treatment is similar for the contagious demands and the vertices for changing vehicles and disinfection activities.

Because some transportation demands need a change of vehicle (of type C or A), the crews have to change their vehicle several times a day at different depots. Because the routes of the crews may evolve dynamically, it is sometimes interesting to change the location of the vehicle changes. The crews are considered in their increasing date of change order, and to each crew is assigned the nearest possible changing depot.

5.5 Decomposition and reconstruction of solutions

At each iteration k , a tabu search (see section 5.8 below) is launched to improve the current solution S . The tabu search is not used for all the routes of a solution, but only for a subset of routes, in order to reduce the computation time. The set of routes S is divided into two subsets denoted S_1 and S_2 , with $S_1 \cup S_2 = S$ ($S_1 \cap S_2$ may be nonempty). This is the *decomposition phase* of a solution.

A solution can be decomposed in two ways: geographically or temporally. At iteration k (line 6),

the route corresponding to the $(k \bmod(|\mathcal{M}|))^{th}$ crew is selected ($|\mathcal{M}|$ is the number of crews). Then, the $|\mathcal{M}| \times 2/3$ routes which are the more similar to this route (geographic or temporal similarity alternatively) are gathered and constitute the set S_1 . The set S_2 is composed by the $|\mathcal{M}| \times 2/3$ routes which are the less similar to the selected route (geographic or temporal similarity alternatively). The tabu search is used on each subset separately and successively, which gives S'_1 and S'_2 .

Reconstructing a solution S' is the *reconstruction phase* of a solution. It consists in making the union of the routes of S'_2 and the routes of S'_1 that are not in S'_2 .

5.6 Diversification

During the search, some solutions can be explored several times (between lines 3 and 15). To avoid this redundancy, a treatment is inserted at line 3 (when building a solution from the adaptive memory) for introducing some diversification. When the same best solution has been generated three times during the K iterations, the selection order of routes in the adaptive memory is inverted. This step turns the exploration to another part of the solution space.

5.7 Post optimization

At the end of the algorithm (line 17), a post-optimization step is applied to the best found solution. This step uses the same process as for the initialization procedure, i.e. a small number of iterations of the tabu search. This optimization phase can be efficient for improving solutions because the tabu search works on all the routes simultaneously.

5.8 Tabu search

The tabu search algorithm [Glover, 1986] and [Hansen, 1986] developed here, has the same structure as in [Taillard *et al.*, 1997]. A solution is a subset of routes from the adaptive memory. The tabu search is performed twice per iteration $k \in \{1, \dots, K\}$. These two launches use the same tabu list. We now describe the major elements of this tabu search algorithm.

The stopping criterion is the maximum number of iterations, defined by $E \times [1 + (k - 1)/F]$ where E and F are two given parameters and k is the iteration number of both the decomposition and the reconstruction phases. This definition is used because after each run of the tabu search, the quality of the initial solution is improved. Thus, the tabu search needs more and more operations to escape from a possibly local optimum.

Algorithm 2 : TS

- 1: **while** the stopping criterion is not met **do**
 - 2: Generate the neighbourhood of the current solution by applying the operator CROSS exchange.
 - 3: Select the best solution, which is not in the tabu list, in this set of neighbors
 - 4: **if** this solution is better than the best known solution **then**
 - 5: Save this solution as the best solution
 - 6: **end if**
 - 7: Update the tabu list
 - 8: **end while**
-

The neighborhood is built with the operator CROSS exchange. This operator is particularly well adapted for a VRP with time windows, but has to be especially re-adapted to our problem (change of vehicle, time window...). It is also a generalization of well-known operators like 2-opt or Or-opt. This operator consists in exchanging two segments belonging to two different routes. It is illustrated in Fig. 6.

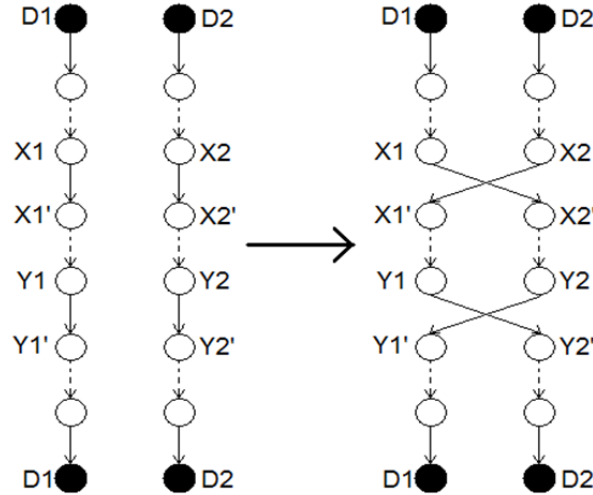


Figure 6: Illustration of CROSS exchange operator

The neighborhood of a solution is obtained by exchanging all sub-segments of all routes of the subset considered leading to a feasible solution. In a tabu search algorithm, one of the most expensive elements in terms of computation time is the exhaustive neighborhood exploration. Four necessary conditions are needed to avoid unfeasible neighbors:

- if $X2'$ and $X1$ correspond to a transportation then $e_{X2'} + \alpha_{X2'} \geq e_{X1} + d(X1, X2')$
- if $X1'$ and $X2$ correspond to a transportation then $e_{X1'} + \alpha_{X1'} \geq e_{X2} + d(X2, X1')$
- if $Y2'$ and $Y1$ correspond to a transportation then $e_{Y2'} + \alpha_{Y2'} \geq e_{Y1} + d(Y1, Y2')$
- if $Y1'$ and $Y2$ correspond to a transportation then $e_{Y1'} + \alpha_{Y1'} \geq e_{Y2} + d(Y2, Y1')$

These conditions check whether the crew of the first route has enough time to go from $X1$ to $X2'$ and from $Y2$ to $Y1'$ (the crew of the second route from $X2$ to $X1'$ and $Y1$ to $Y2'$ respectively). These conditions allow to reduce the neighborhood between two routes. However, a solution obtained after a move is not necessarily valid for a given crew and the feasibility has to be checked (respect of time windows, return of physician on time, availability of vehicles, disinfection, changing of vehicle).

As in [Taillard *et al.*, 1997], the objective value of a solution is saved in the tabu list during a given number of iterations (half the maximum number of iterations of the tabu search). The tabu list contains the objective function values. It is not based on the description of the solutions because the probability for obtaining two different solutions with the same objective value is very low. So, if the objective value of a solution belongs to the tabu list, this solution has most probably already been explored, and thus is considered tabu. Moreover, it is less expensive, in terms of computation time, to store and compare two objective values than two moves or worse, two full solutions. The size of the list is supposed to be sufficiently large for being considered of infinite size.

6 Computational experiments

6.1 Preliminaries

Some formulations exist in the literature for quantifying the degree of dynamism of a problem instance ([Larsen, 2001] and [Ghiani *et al.*, 2003]). We define Ψ :

$$\Psi = \frac{|\mathcal{N}_d|}{|\mathcal{N}|}$$

with \mathcal{N}_d the set of dynamic demands (not known in advance). Ψ does not take into account dynamic demand occurrence times and time windows. That is the reason why Larsen extended the definition of Ψ and introduced Ψ' :

$$\Psi' = \frac{1}{|\mathcal{N}|} \sum_{i \in \mathcal{N}} \frac{T - (e_i + \alpha_i - t_i^{occ})}{T}$$

with $[0, T]$ the planning horizon, and t_i^{occ} the occurrence time of demand i (static demands are such that $t_i^{occ} = 0$). Ψ and Ψ' may vary between 0 (in case of a static problem) and 1 (in case of a pure dynamic problem).

The objective function of the problem is mainly related to the number of subcontracted demands: reducing the number of subcontracted demands globally contributes to reducing the global cost. Furthermore, this is an indicator which is easy to measure and which helps the operators for taking their decisions. An integer linear programming (ILP) model can give a lower bound, denoted LB_{sub} on the number of subcontracted demands. The variables are $x_{ik} = 1$ if team k performs demand i and 0 otherwise ($\forall i \in \mathcal{N}, \forall k \in \mathcal{M}$). Notice that $x_{ik} = 0, \forall k \in \mathcal{M}$ indicates that demand i is subcontracted. The ILP is the following.

- Maximize $\sum_{i \in \mathcal{N}, k \in \mathcal{M}} x_{ik}$
- Subject to:
 - A demand is performed by at most one team: $\forall i \in \mathcal{N}: \sum_{k \in \mathcal{M}} x_{ik} \leq 1$.
 - If the working time of team k does not allow to perform demand i on time, we have:
 $x_{ik} = 0$.
 - If two demands i and j cannot be performed simultaneously on time by the same team:
 $\forall k \in \mathcal{M}: x_{ik} + x_{jk} \leq 1$.

The two last sets of constraints are generated by preprocessing and take account of time windows, travel times and activities of transportation with medical monitoring or contagious transportation. This ILP model allows to determine LB_{sub} for each instance by using CPLEX solver, in less than 30 minutes, 1 minute on average.

6.2 Real-life data set

The information system, used by the operators for managing the transportation demands, contains the data on the demands performed every day. Unfortunately, all the transportation demands and all their information are not systematically entered into the system. Some of them are treated without the help of the information system. Anyway, real-life instances of ten days have been observed. Table 1 gives the degree of dynamism of these instances. Considering the time windows and occurrence times, these instances show a strong feature of dynamism. We note that the average ratios are $\overline{\Psi} = 58\%$ and $\overline{\Psi'} = 64\%$.

| | | | | | | | | | | |
|-----------------|------|------|------|------|------|------|------|------|------|------|
| <i>Days</i> | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| $ \mathcal{N} $ | 125 | 91 | 90 | 92 | 115 | 113 | 128 | 99 | 126 | 132 |
| Ψ | 0,58 | 0,62 | 0,66 | 0,66 | 0,53 | 0,55 | 0,52 | 0,58 | 0,47 | 0,62 |
| Ψ' | 0,72 | 0,74 | 0,75 | 0,76 | 0,68 | 0,69 | 0,70 | 0,69 | 0,62 | 0,73 |

Table 1: Degree of dynamism of ten real-life instances

The real instance concerning the last day has been used for the evaluation of our algorithm. During this day, 132 transportation demands have been performed, with 50 demands known in advance, 6 using medical monitoring, 3 contagious demands and 17 were coming from the EMS. During this day, the operators subcontracted 36 transportation demands and some transportation demands were performed lately.

6.3 Instances generation

Random instances have been generated according to the real activity of the central station. The time period is one day, starting at 6:00 am (time 0) and ending at 9:00 pm.

Data associated to the crews: $|\mathcal{M}| = 11$ crews working per day (timetables in Table 2). The acceptable overtime work is $T = 30$ minutes.

| Crew | 1 | 2 and 3 | 4 | 5 and 6 | 7 and 8 | 9 | 10 and 11 |
|-------|---------|---------|---------|---------|----------|----------|-----------|
| Start | 6:00 am | 7:00 am | 8:00 am | 9:30 am | 12:00 am | 12:30 am | 1:30 pm |
| End | 1:30 pm | 2:30 pm | 3:30 pm | 5:00 pm | 7:30 pm | 8:00 pm | 9:00 pm |

Table 2: Timetables of crews

Data associated to the vehicles: $|\mathcal{K}| = 15$ vehicles with 13 vehicles of type C and 2 of type A .

Data associated to the transportation demands:

- The number of transportation demands per day is 130 (among them 25% to 35% are known in advance). The number of transportation demands with medical monitoring is comprised between 5 and 20 and the number of contagious transportation demands between 2 and 8. The remaining demands are classic.
- Numerical data

$$\begin{array}{lll}
u_i \in [1, 5] & p_i^{care} \in [5, 40] \text{ min.} & \tau_0 = 5 \\
e_i \sim \mathcal{N}(330, 300) \text{ (morning)} & e_i \sim \mathcal{N}(600, 300) \text{ (afternoon)} & \delta = 5 \\
p_{max} = 15 \text{ min.} & p_{ch} = 10 \text{ min.} & p_{phy} = 5 \text{ min.} \\
p_{dis} = 60 \text{ min.} & &
\end{array}$$

- The unexpected demands are known between 10 to 240 minutes before the dates of transportation.

Five sets of 200 instances have been generated, depending on the number of transportation demands coming from the Emergency Medical Service. These demands are known only between 5 and 40 minutes before the effective date of transportation and are very urgent ($u_i = 1$). Five types of instances have been considered: Type I, II, III, IV and V, if 0%, 5%, 10%, 15% and 20% of the 130 demands are coming from the Emergency Medical Service. Our observations show that the ratio of 10% fits best the real activity of the central station. Table 3 shows the measure of the average degrees of dynamism for the different data sets.

| Type of instances | I | II | III | IV | V |
|--------------------|------|------|------|------|------|
| $\overline{\Psi}$ | 0,73 | 0,74 | 0,75 | 0,75 | 0,76 |
| $\overline{\Psi}'$ | 0,77 | 0,78 | 0,79 | 0,80 | 0,81 |

Table 3: Degree of dynamism of the generated instances

The degree of dynamism of the generated instances are higher than for the real data set, because we wanted to introduce the transportation demands that are not captured in the information system of the central station. The ratio of 70% fits best to a real situation.

6.4 Algorithm parameters

All the tests have been performed on a Pentium(R) 4 CPU 3.00 GHz, 1.00 Go of RAM and the programming language used was C++. After some preliminary experiments, performed on 20 instances, the parameters of the algorithm have been set to the following values:

| | |
|---|----------------------|
| Number of random solutions to initialize the adaptive memory | = 50 |
| Size of the adaptive memory ($50 \times \mathcal{M} $) | = 550 routes |
| Number of iterations ($K = 2 \times \mathcal{M} $) | $J = 5$ and $K = 22$ |
| Parameters E and F | $E = 7$ and $F = 3$ |
| Probability selection ρ | = $1/4$ |
| Number of iterations for the TS in the initialization procedure | = 5 |
| Number of iterations for the TS in the post optimization | = 10 |

The algorithm is launched each time an event occurs, so it must quickly find a good solution to help the operators taking a decision. This time is directly related to the number J of iterations of the main loop (line 2) and the value $J = 5$ corresponds to a running time of approximatively 2 seconds per new event.

6.5 Results

The performance and the structure of the algorithm have been tested. In particular, the influence of the use of the adaptive memory and of the impact of the tabu list have been evaluated. We denote by *DynWam* algorithm *Dynamic* Without adaptive memory (in this case, the adaptive memory contains only the best found solution) and *DynWtl* algorithm *Dynamic* Without the tabu list.

These different algorithms are compared by using a relative deviation measure:

$$\Delta(H1, H2) = \frac{f(H1) - f(H2)}{f(H1)}$$

If $\Delta(H1, H2)$ is negative, it means that heuristic $H1$ is lower (better in our case) than heuristic $H2$. Tables 4 and 5 present respectively the results of the comparisons between *Dynamic* and *DynWam* and *DynWtl*. For each type of instance, we note the minimum, average and maximum relative deviation, the standard deviation and $\%DynBest$ which is the percentage of instances where algorithm *Dynamic* returns a better result.

Tables 4 and 5 show that both the adaptive memory and the tabu list have a positive influence on the solution quality. For around 80% and 92,50% of the instances, the solution quality decreases without the adaptive memory and without the tabu list, respectively.

Table 6 presents the comparison between algorithm *Dynamic* and *DynOper*. Even if algorithm *DynOper* seems to be a good method for finding correct solutions, the algorithm *Dynamic* is better in more than 80% of the cases, with a gain of solution cost around 20%.

| Types of instances | $\Delta(Dynamic, DynWam)$ | | | | |
|-----------------------|---------------------------|--------|--------|---------|------------------|
| | min | aver. | max | st.dev. | % <i>DynBest</i> |
| Type I | -48,41% | -6,18% | 3,17% | 5,57% | 92,5% |
| Type II | -23,56% | -4,41% | 11,30% | 5,40% | 89,5% |
| Type III | -30,49% | -4,46% | 4,90% | 4,36% | 88% |
| Type IV | -58,78% | -3,83% | 13,46% | 5,31% | 84,5% |
| Type V | -15,19% | -3,74% | 6,51% | 3,60% | 85% |

Table 4: Comparisons between *Dynamic* and *DynWam*

| Types of instances | $\Delta(Dynamic, DynWtl)$ | | | | |
|-----------------------|---------------------------|--------|--------|---------|------------------|
| | min | aver. | max | st.dev. | % <i>DynBest</i> |
| Type I | -46,45% | -4,71% | 4,66% | 5,23% | 83% |
| Type II | -22,15% | -3,25% | 14,18% | 5,48% | 86% |
| Type III | -20,47% | -3,68% | 4,12% | 3,71% | 84% |
| Type IV | -11,05% | -2,87% | 11,59% | 3,33% | 80% |
| Type V | -11,91% | -2,68% | 7,27% | 3,23% | 80% |

Table 5: Comparisons between *Dynamic* and *DynWtl*

| Types of instances | $\Delta(Dynamic, DynOper)$ | | | | |
|-----------------------|----------------------------|--------|--------|---------|------------------|
| | min | aver. | max | st.dev. | % <i>DynBest</i> |
| Type I | -15,46% | -4,20% | 5,09% | 3,80% | 85% |
| Type II | -16,77% | -3,42% | 12,69% | 5,21% | 81,5% |
| Type III | -14,61% | -3,76% | 10,74% | 3,83% | 85% |
| Type IV | -19,71% | -3,74% | 12,09% | 4,05% | 83% |
| Type V | -18,03% | -4,28% | 6,54% | 3,69% | 88% |

Table 6: Comparisons between *Dynamic* and *DynOper*

Of course, this difference is mainly related to the number of subcontracted demands. We therefore have examined the average number of subcontracted demands obtained by each algorithm. Table 7 presents the results.

| Types | LB_{sub} | <i>Dynamic</i> | <i>DynOper</i> | <i>DynWam</i> | <i>DynWtl</i> |
|----------|------------|----------------|----------------|---------------|---------------|
| Type I | 24 | 34,685 | 36,925 | 36,93 | 37,150 |
| Type II | 26 | 37,465 | 39,345 | 39,495 | 39,785 |
| Type III | 27 | 38,555 | 40,075 | 40,225 | 40,855 |
| Type IV | 31 | 41,86 | 43,145 | 42,935 | 43,725 |
| Type V | 32 | 43,78 | 45,145 | 45,005 | 45,495 |

Table 7: Average number of subcontracted transportation demands

The difference between LB_{sub} and algorithm *Dynamic* is around 10 demands. This difference is explained by the two last constraints of the ILP model. It is not possible to take account all the elements of the problem (time of disinfection, change of vehicle, etc..). Anyway, the algorithm finds on average two subcontracted demands less than the other algorithms, which represents an interesting profit for the hospital.

For the real data set, we have executed algorithm *Dynamic*. for this instance, the algorithm finds a solution with 33 subcontracted demands, whereas a solution with 37 subcontracted demands is found by *DynOper*. Among the 82 demands that were know in real time, 43 can be inserted directly to the current solution by algorithm *Dynamic*.

7 Conclusion

We have considered a dynamic transportation problem of patients between care units in a French hospital complex. The objective function was to minimize the total transportation cost. A tabu search algorithm has been developed. The algorithm supports an adaptive memory which stores routes and which consists in performing a tabu search algorithm iteratively: for improving the set of initial solutions, for the neighbourhood exploration and finally for improving the final solution. This method presents the advantage of *flexibility* and *robustness*: flexibility because it can easily be adapted to take pre-assignments into account, in order to respect the choices of operators; robustness because if the current solution becomes unfeasible due to unexpected events (like emergency demands), a new good solution can be quickly found, with the help of the adaptive memory.

Our results show that the proposed method can quickly find good solutions and can provide an help to the operator in real time for assigning vehicles to new transportation demands.

One future research direction is to consider that new unexpected events may occur like vehicle breakdowns or that transportation durations are not constant but traffic dependent.

Acknowledgments

The authors would like to thank the anonymous reviewers for their thoughtful comments and suggestions that improved the paper.

References

- [Alsalloum and Rand, 2006] Alsalloum OI and Rand GK. Extensions to emergency vehicle location models. *Computers & Operations Research*, 2006, 33(9): 2725–2743.
- [Attanasio *et al.*, 2004] Attanasio A, Cordeau J-F, Ghiani G, Laporte G. Parallel tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem. *Parallel Computing*, 2004, 30(3): 377–387.
- [Araz *et al.*, 2007] Araz C, Selim H, Ozkarahan I. A fuzzy multi-objective covering-based vehicle location model for emergency services. *Computers & Operations Research*, 2007, 34(3): 705–726.
- [Badeau *et al.*, 1997] Badeau P, Gendreau M, Guertin F, Potvin J-Y, Taillard E. A parallel tabu search heuristic for the vehicle routing problem with time windows. *Parallel Computing in Transport Research*, 1997, 5(2): 109–122.
- [Beaudry *et al.*, 2008] Beaudry A, Laporte G, Melo T, Nickel S. Dynamic transportation of patients in hospitals. *OR Spectrum*, 2008, 32(1): 77–107.
- [Berbeglia *et al.*, 2010] Berbeglia G, Cordeau J-F, Laporte G. Dynamic Pickup and Delivery Problems. *European Journal of Operational Research*, 2010, 202(1): 8–15.
- [Borndörfer *et al.*, 1997] Borndörfer R, Grötschel M, Klostermeier F, Küttner C. Telebus Berlin: vehicle scheduling in a dial-a-ride system. Technical report SC 97–23, Konrad-Zuse-Zentrum für Informationstechnik, Berlin, 1997.

- [Cordeau and Laporte, 2003] Cordeau J-F, Laporte G. A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Transportation Research Part B: Methodological*, 2003, 37(6): 579–594.
- [Cordeau and Laporte, 2007] Cordeau J-F, Laporte G. The dial-a-ride problem: models and algorithms. *Annals of Operations Research*, 2007, 153(1): 29–46.
- [Coslovich *et al.*, 2005] Coslovich L, Pesenti R, Ukovich W. A two-phase insertion technique of unexpected customers for a dynamic dial-a-ride problem. *European Journal of Operational Research*, 2005, 175(3): 1605–1615.
- [Desaulniers *et al.*, 2002] Desaulniers G, Desrosiers J, Erdmann A, Solomon MM, Soumis F. The VRP with pickup and delivery. In: Toth P, Vigo D (Eds.) *The vehicle routing problem. SIAM Monographs on Discrete Mathematics and Applications*, Philadelphia, 2002, pp. 225–242.
- [Desrosiers *et al.*, 1991] Desrosiers J, Dumas Y, Soumis F, Taillefer S, Villeneuve D. An algorithm for mini-clustering in handicapped transport. Cahiers du GERARD Report G-91-02, École des Hautes Études Commerciales, Montréal, Canada, 1991.
- [Dumas *et al.*, 1989] Dumas Y, Desrosiers J, Soumis F. Large scale multi-vehicle dial-a-ride problems. Cahiers du GERARD Report G-89-30, École des Hautes Études Commerciales, Montréal, Canada, 1989.
- [Gendreau *et al.*, 1999] Gendreau M, Guertin F, Potvin J-Y, Taillard E. Parallel tabu search for Real-Time Vehicle Routing and Dispatching. *Transportation Science*, 1999, 33(4): 381–390.
- [Gendreau *et al.*, 2006] Gendreau M, Guertin F, Potvin J-Y, Séguin R. Neighborhood search heuristics for a dynamic vehicle dispatching problem with pick-ups and deliveries. *Transportation Research Part C: Emerging Technologies*, 2006, 14(3): 157–174.
- [Ghiani *et al.*, 2003] Ghiani G, Guerriero F, Laporte G, Musmanno R. Real-time vehicle routing: Solution concepts, algorithms and parallel computing strategies. *European Journal of Operational Research*, 2003, 151(1): 1–11.
- [Glover, 1986] Glover F. Future Paths for Integer Programming and Links to Artificial Intelligence. *Computers & Operations Research*, 1986, 13(5): 533–549.

- [Hansen, 1986] Hansen P. The steepest ascent mildest descent heuristic for combinatorial programming. Congress on Numerical Methods in Combinatorial Optimization, Capri, 1986.
- [Hanne *et al.*, 2009] Hanne T, Melo T, Nickel S. Bringing Robustness to Patient Flow Management Through Optimized Patient Transports in Hospitals. *Interfaces*, 2009, 39(3): 241–255.
- [Horn *et al.*, 2002] Horn MET. Fleet scheduling and dispatching for demand-responsive passenger services. *Transportation Research Part C: Emerging Technologies*, 2002, 10(1): 35–63.
- [Ichoua *et al.*, 2000] Ichoua S, Gendreau M, Potvin J-Y. Diversion issues in real-time vehicle dispatching. *Transportation Science*, 2000; 34(4): 426–438.
- [Ichoua *et al.*, 2007] Ichoua S, Gendreau M, Potvin J-Y. Planned route optimization for real-time vehicle routing. In: Zeimpekis V, Tarantilis CD, Giaglis GM and Minis I (Eds), *Dynamic fleet management: concepts, systems, algorithms and case studies*. Springer Science + Business Media, New York, États-Unis, 2007. p. 1-18.
- [Ioachim *et al.*, 1995] Ioachim I, Desrosiers J, Dumas Y, Solomon MM, Villeneuve D. A request clustering algorithm for door to door handicapped transportation. *Transportation Science*, 1995, 29(1): 63–78.
- [Jaw *et al.*, 1986] Jaw JJ, Odoni AR, Psaraftis HN, Wilson NHM. A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows. *Transportation Research Part B: Methodological*, 1986, 20(3): 243–257.
- [Kergosien *et al.*, 2008] Kergosien Y, Lenté C, Billaut J-C. A dynamic transportation problem of patients between care units. 34rd International Conference on Operational Research Applied to Health Services (ORAHS’2008), Toronto, Canada, 2008.
- [Laporte, 1997] Laporte G. Modeling and solving several classes of arc routing problems as traveling salesman problems. *Computers & Operations Research*, 1997, 24(11): 1057–1061.
- [Larsen, 2001] Larsen A. The dynamic vehicle routing problem. Ph.D. Thesis, Institute of Mathematical Modeling, Technical University of Denmark, 2001.
- [Madsen *et al.*, 1995] Madsen OBG, Ravn HF, Rygaard JM. A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives. *Annals of Operations Research*, 1995, 60(1): 193–208.

- [Melachrinoudis *et al.*, 2007] Melachrinoudis E, Ilhan AB, Min H. A dial-a-ride problem for client transportation in a health-care organization. *Computers & Operations Research*, 2007, 34(3): 742–759.
- [Psaraftis, 1980] Psaraftis HN. A dynamic programming approach to the single-vehicle, many-to-many immediate request dial-a-ride problem. *Transportation Science*, 1980, 14(2): 130–154.
- [Rajagopalan *et al.*, 2008] Rajagopalan HK, Saydam C, Xiao J. A multiperiod set covering location model for dynamic redeployment of ambulances. *Computers & Operations Research*, 2008, 35(3): 814–826.
- [Shang and Carolyn, 1996] Shang, JS, Carolyn KC. Multicriteria pickup and delivery problem with transfer opportunity. *Computers and Industrial Engineering*, 1996, 30(4): 631–645.
- [Stein, 1978] Stein D. An asymptotic, probabilistic analysis of a routing problem. *Mathematics of Operations Research*, 1978, 3(2): 89–101.
- [Taillard *et al.*, 1997] Taillard E, Badeau P, Gendreau M, Guertin F, Potvin J-Y. A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation science*, 1997, 31(2): 170–186.
- [Teodorovic and Radivojevic, 1998] Teodorovic D, Radivojevic G. A fuzzy approach to dynamic Dial-A-Ride problem. *Fuzzy Sets and Systems*, 1998, 116(1): 22–33.
- [Xiang *et al.*, 2006] Xiang Z, Chu C, Chen H. A fast heuristic for solving a large-scale static dial-a-ride problem under complex constraints. *European Journal of Operational Research*, 2006, 174(2): 1117–1139.
- [Xiang *et al.*, 2008] Xiang Z, Chu C, Chen H. The study of a dynamic dial a ride problem under time dependant and stochastic environments. *European Journal of Operational Research*, 2008, 185(2): 53–551.